Advanced Perl Programming

Course Summary

Description

This course provides advanced Perl coding tools that go beyond the basics. For the programmer that has mastered the basic building blocks of Perl, this course focuses on several common application domains (e.g. Networking, database access) . Perl best practices are also emphasized.

Course Objectives

After completing this course, students will be able to

- Create efficient data structures using references
- Understand advanced text parsing techniques
- Access data from relational database systems
- Create reusable modules
- Design and use objects
- Create simple TCP/IP clients and servers
- Parse XML files
- Implement unit tests for classes and modules
- Write a GUI front end for a module or framework

Topics

- Shortcuts
- Text Parsing
- References
- Using CPAN to access the Perl Library
- Creating Reusable Modules
- Objects in Perl
- Exception Handling
- Database Access
- Network Programming
- XML manipulation
- Checking, testing, and distributing code
- Developing GUI applications

Audience

This course is designed for programmers who want to leverage their basic Perl knowledge into production-quality tools.

Prerequisites

Students should have a solid grasp of basic Perl programming.

Duration

Four days

Advanced Perl Programming

Course Outline

I. References
- A. Creating references
- B. Anonymous arrays and hashes
- C. Dereferencing
- D. References and subroutines
- E. References and arrays
- F. Complex data structures

II. Data wrangling
- A. Reading text files
- B. Creative use of <> and $/
- C. Matching and substituting
- D. RE review
- E. Using backreferences
- F. Parsing lines
- G. Using here documents and __END__
- H. Converting data with pack/unpack

III. Shortcuts and defaults
- A. The ubiquitous $_
- B. How to use <>
- C. Pattern matching in brief
- D. File tests on _
- E. Command-line shortcuts

IV. Modern Perl
- A. New features since 5.0
- B. Omitting parentheses
- C. Using say()
- D. Higher-order functions
- E. Smart matching

V. Using Modules
- A. A quick tour
- B. Use vs. require
- C. Library files
- D. Perl modules
- E. Bundled libraries and modules
- F. Case study: Using Getopt::Long

VI. Creating Modules
- A. Review of subroutines
- B. Understanding my () and local ()
- C. Packages and the symbol table
- D. Mechanics of module creation
- E. Exporting subroutines and data
- F. Using BEGIN and END
- G. Good module design

VII. Classes
- A. Perl's approach to OOP
- B. Indirect subroutine call syntax
- C. OOP Terminology and Perl
- D. Constructors
- E. Data structures
- F. Attributes and methods
- G. Using objects

VIII. Exception handling
- A. Simple exception handling
- B. About eval
- C. Run-time eval
- D. Compile-time eval
- E. Raising error messages

IX. Database Access
- A. Understanding the DBI
- B. Connecting to a database
- C. Executing queries and fetching results
- D. Obtaining metadata
- E. Advanced DBI issues

X. Graphics programming with Tk
- A. About Tk
- B. A basic GUI app
- C. Widgets
- D. Fonts and colors
- E. Data entries
- F. Event handling
- G. Arranging widgets

XI. Network Programming
- A. About clients and servers
- B. Ports and IP addresses
- C. The IO::Socket module
- D. A simple client
- E. Accessing standard services
- F. A simple server
- G. Serializing data structures with freeze/thaw

XII. Threads
- A. About threads
- B. Using the threads module
- C. Waiting for exit
- D. Sharing data

Advanced Perl Programming

Course Outline (cont'd)

XIII. Examining XML
- A. SAX vs DOM
- B. Which module does what?
- C. Introducing XML::Twig
- D. Searching for tags
- E. Creating and manipulating tags
- F. Parsing SAX-style

XIV. Enterprise development
- A. Coding standards
- B. Programmer tools
- C. Module::Starter